Real-Time Object Recognition Based on Cortical Multi-scale Keypoints

Kasim Terzić, J.M.F. Rodrigues and J.M.H. du Buf

Vision Laboratory (LARSyS), University of the Algarve, Faro, Portugal
{kterzic,jrodrig,dubuf}@ualg.pt

Abstract. In recent years, a large number of impressive object categorisation algorithms have surfaced, both computational and biologically motivated. While results on standardised benchmarks are impressive, very few of the best-performing algorithms took run-time performance into account, rendering most of them useless for real-time active vision scenarios such as cognitive robots. In this paper, we combine cortical keypoints based on primate area V1 with a state-of-the-art nearest neighbour classifier, and show that such a system can approach state-of-the-art categorisation performance while meeting the real-time constraint.

1 Introduction

Object recognition and categorisation has seen tremendous progress in recent years. As noted by Boiman et al. [1], in a few years the best-performing methods went from scoring around 20% on Caltech 101 to almost 90%. This has led to newer, more challenging datasets like Caltech 256 [2] and PASCAL VOC [3], with similar progress. These improvements went hand-in-hand with increased computational power and improvements in machine learning methods, allowing for learning very complex relationships from training images. But while this research has pushed the boundaries of what is possible, most of the best-performing methods are very slow. While directly comparing reported runtimes is difficult due to differences in implementation details and different hardware used, most authors who report how long it takes to finish processing a benchmark like Caltech 101 mention hours and days. Even with more powerful computers, most of these methods will not be usable on mobile robots for years.

What is the cause of this slowness? The majority of approaches are based on machine learning methods [4, 5]. Images (or parts of images) are described by feature vectors, which are often quantised into a codebook, and a set of classifiers is learned from a training set. With these approaches, most of the time is spent in the learning stage, which is usually only done once but it can take a long time. Once the category models have been learned, recognition is usually fast. However, learning new classes typically requires re-learning all the classifiers, making these methods less suitable for applications such as cognitive robotics – a robot should not have to wait many hours to learn a new class.

A second group of approaches do not do any training, but compare descriptors from a query image directly with all the descriptors from all the labelled model

images. It has recently been shown that nearest neighbour classifiers can match the performance of machine-learning classifiers on several challenging benchmarks, and that they effectively approximate a *maximum a posteriori* classifier [1]. Given their ability to learn new categories instantly, this makes them an attractive option for cognitive robotics, but the absence of a training stage is countered by increased computational complexity during recognition. In [1], the authors sample descriptors on a dense grid, leading to reported recognition times of about 1.6 seconds per class per image, so almost 3 minutes per image for the Caltech 101 dataset. This too is not acceptable for a cognitive robot.

In this paper, we combine the nearest neighbour classifier from [1] with the cortical keypoints from [6] and propose a number of practical optimisations which drastically reduce the required processing. While this efficient method doesn't match the classification performance of today's best algorithms, it manages to match some recent state-of-the-art methods, all while achieving real-time performance.

1.1 Related Work

In recent years many methods have performed well in object recognition and categorisation tasks. The majority of them extract feature vectors from images and use a powerful classifier to discriminate between classes. Support Vector Machines are most commonly used [7, 8, 2, 5, 9, 10], but also other types of classifiers, such as random forests [11] and Non-linear Kernel Discriminant Analysis (NKDA) [4]. These approaches feature an expensive learning stage, after which recogntion is typically fast. Often improvements come from a proper selection of the SVM kernel [8, 9] or the selection of a proper vector quantisation step.

A second approach to category-level recognition uses non-parametric classifiers, typically nearest-neighbour methods [1, 5, 12]. Nearest-neighbour search is usually done using K-D trees, making the problem computationally tractable. Traditionally, they measure a total image-to-image distance, but recently it has been shown that image-to-class is a more effective measure [1]. These methods do not require a learning stage, but since they typically store many labelled descriptors, recognition is slow.

Features representing an image are often extracted on a grid [5, 1, 13], but some methods are based on interest point detectors, such as Harris corners [4], Difference-of-Gaussians [14] or others [15] in order to first locate most important regions in the image, but the selection of the right type of interest points is crucial. Some detectors detect very general features whereas others detect highly object-specific ones, and a trade-off is needed for good categorisation performance.

In this paper, we propose using biologically-inspired multi-scale keypoints for capturing the most significant regions in the image, and we show that using these types of keypoints can significantly speed up object categorisation.

Algorithm 1 Our real-time classification algorithm

Select scales $\Lambda \leftarrow \{\lambda_{max} \dots \lambda_{min}\}$
$\forall \lambda \in \Lambda$ extract descriptors $D^\lambda \leftarrow \{d_1^\lambda \dots d_N^\lambda\}$ at keypoint locations
$C \leftarrow \{c_1 \dots c_M\}$
$\forall c \in C \quad \text{dist}_C \leftarrow 0$
for all $\lambda \in \Lambda$ **do**
    $\forall d_i^\lambda \in D^\lambda \quad \forall c \in C \quad \text{dist}_c \leftarrow \text{dist}_c + \sum_{i=1}^{N} \|d_i^\lambda - \text{NN}_C(d_i^\lambda)\|^2$
    if $\text{dist}_{c_i} > 2\text{dist}_C^{\min}$ **then**
        $C \leftarrow C \setminus \{c_i\}$
    end if
end for
$\hat{C} = arg\min_C \ \text{dist}_C$

2 Method

Our method starts by resizing the image to 300x300 pixels, followed by keypoint extraction. We use cortical keypoints from [6], which detect meaningful events both at fine scales (corners, junctions) and at coarse scales (blobs). The choice of the keypoint detection algorithm is important: we have experimented with other interest point detectors, such as DoG (SIFT) and DoH (SURF), but they did not work as well in our categorisation experiments. The keypoints are extracted by a series of filtering operations: simple cells (8 oriented odd and even Gabor wavelets with wavelength λ), complex cells (modulus of the simple cells in quadrature), followed by double-stopped cells and two inhibition schemes (see [6] for details). We extract keypoints at wavelengths $\lambda \in \{11 \dots 64\}$, with λ spaced half an octave apart (6 scales in total). This yields a nice balance between detail and coarse image features.

We then extract a SIFT descriptor at each keypoint location. The width of the SIFT descriptor is set to 2λ. Descriptors smaller than 16×16 show little discriminance between classes. The SIFT descriptor is used to facilitate comparisons to similar work, but we are planning to explore more biologically plausible descriptors in the future. The "learning" stage consists of extracting all keypoints and their descriptors from all labelled images (*training* images) and storing them in class-specific K-D trees to enable a fast nearest-neighbour search.

For each query image (*testing* image), we extract keypoints and descriptors as before. For each descriptor $d_i \in \{d_1 \dots d_N\}$, we find the nearest neighbour (in descriptor space) $\text{NN}_C(d_i)$ in each class C and sum all distances per class, as in [1]:

$$\text{dist}_C = \sum_{i=1}^{N} \|d_i - \text{NN}_C(d_i)\|^2. \tag{1}$$

The winning class is the class which minimises the total distance

$$\hat{C} = arg\min_C \ \text{dist}_C. \tag{2}$$

We note that, like in [1], we are grouping descriptors from labelled images *per class*, and not per image.

Focusing attention on keypoints significantly reduces the number of nearest-neighbour lookups needed. Most nearest-neighbour approaches extract descriptors from a densely sampled grid, which can lead to about 90,000 descriptors per image. Using Caltech 101, which has 101 categories, typically with 30 images used for training, categorising a single image requires about 2.5×10^{13} nearest neighbour lookups. It is easy to see why recognition is slow, despite using efficient nearest neighbour lookups. Our keypoint extraction stage typically extracts about 500 keypoints per image, reducing the number of lookups to about 7.5×10^8, a speedup of $30000\times$ in the case of full, exhaustive search! In practice, the use of K-D trees reduces the nearest-neighbour search to logarithmic time, so the speedup is smaller (roughly linear with the reduction of descriptors per image), but it is still significant. We pay for this speedup by a drop in classification rate, as will be seen in the following section, but the drop is acceptable for many applications.

2.1 Real-Time Active Vision

Since 7.5×10^8 is still a large number, we apply a coarse-to-fine strategy motivated by biological vision. Most keypoints are extracted at fine scale, capturing fine detail, and relatively few are detected at coarse scales, capturing larger structures. Because of this, the matching of all keypoints at coarse scales is less expensive. We therefore begin by processing keypoints at coarse scales ($\lambda \geq 32$), after which we discard all classes whose total distance dist_C is more than twice the minimum distance $\text{dist}_C^{\min}$. We keep doing this at each finer scale, so descriptors at the finest scales are only matched for a small number of classes which look promising. With many classes and many labelled images to compare to, this can cut the processing time by a further factor of 10, without negatively affecting the classification rate.

The most attractive property of our method is that further speedups are possible: one can restrict the number of scales used, which will speed up the processing, but hurt performance. This kind of trade-off is inherent in active vision, such as biological vision and vision for cognitive robots, where the amount of processing is task- and context-specific. Using our method, a suitable trade-off can be chosen during recognition, depending on the number of keypoints which can be processed in the available time. This is not the case with many learning-based methods, where the feature vector describing the image has to match the feature vectors used during training. With our method, an agent can choose to only process very coarse scales (so feature extraction and matching are fast) and sacrifice performance, if speed is required. Or it can choose to spend more time in order to obtain more accurate results, by using more scales. This way a range of speeds is possible, ranging from processing the entire Caltech 101 dataset in a few seconds (using only keypoints at the coarsest scale), to taking several minutes to complete the same job. If one samples enough descriptors, the performance would match the original grid approach from [1], but it would take days.

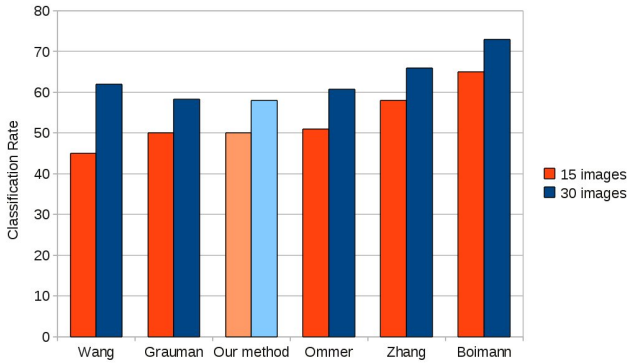

Fig. 1. Comparison with several state-of-the-art algorithms from recent years on the Caltech 101 dataset, using 15 and 30 training images and one type of descriptor (SIFT in our method). We compared against Wang et al. [16], Grauman and Darrell [7], Ommer and Buhmann [4], Zhang at al. [12] and Boimann et al. [1].

3 Evaluation

We have evaluated our nearest-neighbour method on object recognition (COIL-100 dataset [17]) and object categorisation (Caltech 101 dataset [18]) tasks. We have primarily compared against the state-of-the-art nearest-neighbour approach of [1] which we use as a baseline in terms of classification and runtime performance, but have also included some other recent state-of-the-art approaches in the comparison. In these tests, the 130-dimensional descriptors d_i consist of a 128-dimensional SIFT descriptor and scaled x and y image coordinates of the descriptor (using scaling factor $\alpha = 3$).

We first looked at object recognition on the COIL-100 dataset. We used 6 training views and 66 testing views for each class. Our performance is slightly lower than [1] (93.5% vs. 95.5%) but considerably faster, classifying 6600 images in 2'44" vs. 122 minutes, or 40 vs. 0.9 frames per second. For the baseline, we used a grid of descriptors 3 pixels apart; a more densely spaced grid would improve the performance but further increase the time difference. Please note that in this comparison we only measured matching time, not feature extraction, although our approach is also faster there.

We used the Caltech 101 dataset for testing categorisation performance. We emphasise that we only compare against methods which use a single descriptor, not against those which combine many different features. As expected, our algorithm is fast, managing to process the entire Caltech 101 dataset (about 6000 testing images compared against 3000 models) in about 30 minutes, including feature extraction. The classification itself took only 15 minutes with 6 scales and 30 labelled images per class, i.e. feature extraction and matching are balanced in terms of CPU time. As can be seen from Fig. 1, our results are lower than those of state-of-the-art methods [12, 1], but are competitive with some

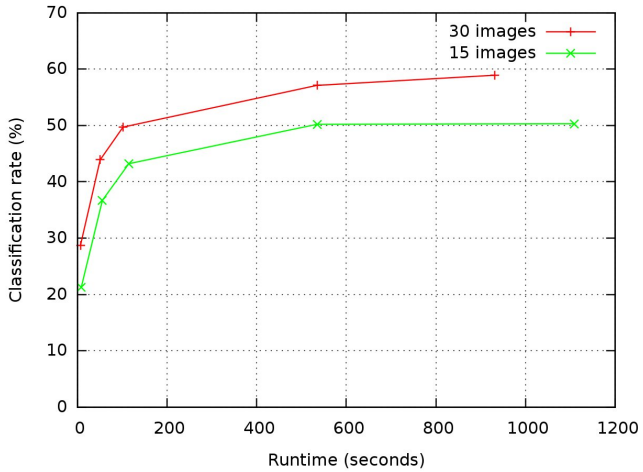

Fig. 2. Runtime vs. accuracy trade-off. Our method can classify all the images from the Caltech 101 dataset in about 7 seconds, and additional processing time gradually improves performance. Note that the time is measured in seconds as opposed to hours.

methods from only a few years ago, like [16, 7, 4]. We achieve 50.2% and 58.9% with 15 and 30 labelled images, respectively. At the same time, our method is much faster than all the listed methods. We note that the multi-scale coarse-to-fine approach did not negatively effect the classification rate in our experiments, same results were obtained when all points were used. It should be emphasised that the two best-performing methods in this comparison both use grids, which are a bit adapted to special properties of the Caltech dataset (see the discussion in [13]), which improves their performance. Also, the approach from [4] uses a very complex hierarchical model, and we can match its performance by using a much simpler approach which requires only a fraction of the time.

Fig. 2 shows the effect of runtime on the classification rate on the Caltech 101 dataset. At one extreme, using only the coarsest scales leads to a classification rate of only 20%, but the entire dataset is processed in only a few seconds. At the other extreme, we reach a competitive classification rate of 59%, using all available scales. We could extend our feature set by adding more descriptors until they cover the entire grid as used by [1], and we would match both their performance and their runtime. Using the standard parameters used throughout this paper (6 scales, coarse-to-fine strategy), we get a compromise: a classification rate 10-15 percentage points below the state of the art, but orders of magnitude faster than competing methods. An active agent can choose how much processing time to invest in a classification task, and get an appropriate classification rate.

4 Discussion

Evaluation shows that our keypoint-based method significantly outperforms the nearest-neighbour baseline in terms of runtime, achieving real-time performance on a standard i5 quad-core processor. At the same time, classification performance is somewhat lower. Even so, it is amazing that it can compete with some older methods, which took many hours for training and optimising non-linear kernels to reach such a high performance. We find this trade-off acceptable: many applications only feature a relatively small number of classes, but absolutely require fast learning and real-time recognition. For such applications, most state-of-the-art approaches are unsuitable, and our method clearly fills this gap.

The most interesting property of the presented algorithm is that the runtime vs. accuracy trade-off is a fluent one, and can be selected at any time during recognition. This allows for real-time and task-driven vision, where limited resources (such as on a cognitive mobile robot) can be optimally allocated to solve a wide range of problems – from rough but fast landmark detection during navigation, to close inspection of scene objects before performing complex tasks. Given enough time, the performance of our algorithm approaches that of the best nearest-neighbour algorithms, which are known to be among the best-performing categorisation methods.

5 Conclusions

We presented a very fast object categorisation algorithm which is orders of magnitude faster than all state-of-the-art categorisation algorithms known to us. Although the classification performance matches that of algorithms from a few years ago, it is not competitive with today's state of the art. However, its speed and ability to choose the accuracy vs. performance trade-off during recognition make it extremely attractive for cognitive robotics and other applications where runtime is an issue.

Our work shows that world-class categorisation performance need not be slow. Although there is still a significant gap in classification rate between our work and best-performing methods, the difference in computational complexity is huge while the results are sufficient for many applications. We believe that this work is only the beginning, and that further tweaks can bring performance closer to the leading methods, while still maintaining real-time performance.

We are currently working on a more biologically plausible model, which replaces SIFT descriptors by descriptors based on simple and complex cell responses. We are also exploring hierarchical representations, using co-occurrences of features and feature clusters to improve classification performance.

Acknowledgements. This work was supported by the EU under the grant ICT-2009.2.1-270247 *NeuralDynamics* and the Portuguese FCT under the grant PEst-OE/EEI/LA0009/2011.

References

1. Boiman, O., Shechtman, E., Irani, M.: In Defense of Nearest-Neighbor Based Image Classification. In: Proc. CVPR, Anchorage (2008)
2. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology (2007)
3. Everingham, M.: The VOC 2006 database (2006) http://pascallin.ecs.soton.ac.uk/challenges/VOC/databases.html [Online; accessed 20-Jan-2010].
4. Ommer, B., Buhmann, J.: Learning the compositional nature of visual object categories for recognition. IEEE T-PAMI **32** (2010) 501–516
5. Varma, M., Ray, D.: Learning The Discriminative Power-Invariance Trade-Off. In: Proc ICCV, Rio de Janeiro (2007) 1–8
6. Rodrigues, J., du Buf, J.: Multi-scale keypoints in V1 and beyond: Object segregation, scale selection, saliency maps and face detection. BioSystems **86** (2006) 75–90
7. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features. In: Proc. ICCV, Beijing (2005) 1458–1465
8. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: Proc. CIVR, Amsterdam (2007) 401–408
9. Zhang, J., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: a comprehensive study. IJCV **73** (2007) 213–238
10. Kumar, A., Sminchisescu, C.: Support Kernel Machines for Object Recognition. In: Proc. ICCV, Rio de Janeiro (2007)
11. Bosch, A., Zisserman, A., Munoz, X.: Image Classification using Random Forests and Ferns. In: Proc. ICCV, Rio de Janeiro, Brazil (2007)
12. Zhang, H., Berg, A.C., Maire, M., Malik, J.: SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In: Proc. CVPR, New York (2006) 2126–2136
13. Pinto, N., Cox, D.D., dicarlo, J.J.: Why is Real-World Visual Object Recognition Hard? PLOS computational biology **4** (2008) 0151–0156
14. Lowe, D.: Object recognition from local scale-invariant features. In: Proc. ICCV. (1999) 1150–1157
15. Mikolajczyk, K., Leibe, B., Schiele, B.: Local features for object class recognition. In: Proc. ICCV, Washington DC, IEEE Computer Society (2005) 1792–1799
16. Wang, G., Zhang, Y., Fei-Fei, L.: Using dependent regions for object categorization in a generative framework. In: Proc. CVPR, New York (2006) 1597–1604
17. Nene, S.A., Nayar, S.K., Murase, H.: Columbia Object Image Library (COIL-100) (1996)
18. L. Fei-Fei, R.F., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: Proc. CVPR Workshop on Generative-Model Based Vision, Washington DC (2004)